MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1015 January, 1988

A LEXICAL CONCEPTUAL APPROACH TO GENERATION FOR MACHINE TRANSLATION

Bonnie J. Dorr

ABSTRACT: Current approaches to generation for machine translation make use of direct-replacement templates, large grammars, and knowledge-based inferencing techniques. Not only are rules language-specific, but they are too simplistic to handle sentences that exhibit more complex phenomena. Furthermore, these systems are not easily extendable to other languages because the rules that map the internal representation to the surface form are entirely dependent on both the domain of the system and the language being generated. Finally an adequate interlingual representation has not yet been discovered; thus, knowledge-based inferencing is necessary and syntactic cross-linguistic generalization cannot be exploited.

This report introduces a plan for the development of a theoretically based computational scheme of natural language generation for a translation system. The emphasis of the project is the mapping from the lexical conceptual structure of sentences to an underlying or "base" syntactic structure called *deep* structure. This approach tackles the problems of *thematic* and *structural* divergence, *i.e.*, it allows generation of target language sentences that are not thematically or structurally equivalent to their conceptually equivalent source language counterparts. Two other more secondary tasks, construction of a dictionary and mapping from deep structure to surface structure, will also be discussed.

The generator operates on a constrained grammatical theory rather than on a set of surface level transformations. If the endeavor succeeds, there will no longer be a need for large, detailed grammars; general knowledge-based inferencing will not be necessary; lexical selection and syntactic realization will be facilitated; and the model will be general enough for extension to other languages.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research has been provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124, and also in part by NSF Grant DCR-85552543 under a Presidential Young Investigator's Award to Professor Robert C. Berwick. Useful guidance and commentary were provided by Bob Berwick, Ed Barton, Michael Brent, Bruce Dawson, Sandiway Fong, and Michael Kashket. This report is an extended version of a Ph.D. proposal submitted in November, 1987.

1 Introduction

This report introduces a new scheme for natural language generation based on *lexical conceptual structure*, which represents meaning through predicate decomposition.[1] For example, the word *capture* would be represented as:[2]

```
(event CAUSE (thing X) (poss BE (thing X) (thing Y)) (property FORCEFULLY))
```

In other words, *capture* is viewed as an event in which an agent (X) forcefully causes a theme (Y) to be possessed by the agent.

The goal of the project is to produce a language-independent system suitable for a generation component of a machine translator. Lexical conceptual structure is used to ease the complicated operations associated with generation, lexical selection, and syntactic realization. In particular, these operations are difficult when semantically equivalent source and target language verbs are not thematically or structurally equivalent. This situation is usually apparent when there is a choice between two or more target language translations. For example, the English word *slash* might be translated as the Spanish word *cortar* (literally, *to cut*), or the composite Spanish form *dar cuchilladas a* (literally, *give knife-wounds to*). The correct lexical selection and syntactic realization of the surface form in such cases is based on a systematic mapping between the lexical conceptual forms of the source and target languages. This will be discussed in more detail in section 4.3.

Previously, generation systems did not provide a representation of "meaning" for the verbs being generated; rather, language dependent templates, inferencing procedures, and network searching rules were used to select the target language verb. Such systems did not take predicate-argument structures into account; thus, they could not explain thematic or structural divergence. Furthermore, cross-linguistic generalization was ignored since the templates and networks were specifically tailored to the languages handled by the system. The approach described here does not make use of language-dependent devices found in older systems. Instead, verbs are defined in terms of many semantic components that contribute to the overall meaning; these composite structures can then be mapped cross-linguistically in order to arrive at target language forms.

The next five sections describe the generation system. The second section provides the background for natural language generation in machine translation. First, a brief description of the theory behind the generation scheme is given. Then existing generation schemes will be discussed and their shortcomings will be addressed.

A plan for the development of a theoretically based computational scheme will be introduced in the third section. Three components of the system will be identified: (1) the dictionary; (2) the morphological/syntactic synthesizer; and (3) the module that maps lexical conceptual structure to deep structure. The third component is the emphasis of this discussion.

---

[1]The representation adopted here is as formulated by Hale and Laughren (1983), and Hale and Keyser (1986).

[2]The modifiers *posit*, *poss* and *ident* stand for *positional*, *possessional* and *identificational* respectively. Words in upper-case are the primitive units of meaning.

How the scheme embodies linguistic theory will be explained in the fourth section. Examples of problems that might be encountered during generation of English and Spanish will be presented. Finally, the goals of the scheme will be described.

The fifth section presents a description of the work that needs to be done in order to accommodate the scheme. The generator will replace the generation component that is currently part of the UNITRAN machine translation system.[3] The new generator will operate on a constrained grammatical theory rather than on a set of surface level transformations. The basic building blocks of the system will be discussed. Also, methods of testing and evaluating the system will be presented.

In the sixth section, some of the difficulties that might arise in the development of the scheme are addressed.

2 Background for Research

This section introduces the background for a generation scheme based on lexical conceptual structure, and provides a description of three other commonly used generation designs: (1) direct-replacement, (2) syntactic-based generation and (3) semantic-based generation. The advantages and disadvantages of these three designs will be discussed. Finally, it will present arguments for why a design based on lexical conceptual structure is an improvement over other designs.

2.1 Lexical Conceptual Structure Approach to Generation

The work of Jackendoff (1972) has influenced much of the lexical-semantic work of the Lexicon project at MIT. The representation adopted is *lexical conceptual structure* (henceforth LCS). According to Rappaport and Levin (1986), LCS encodes a verb's meaning through predicate decomposition. For example, the LCS for the word *put* is:

```
(event (posit MOVE (thing X) (thing Y) (place Z)))
```

Linking rules relate variables in the LCS to the variables in the *predicate-argument structures*, which provide an explicit representation of hierarchical relations between the verb and its arguments. For example, the predicate-argument structure for *put* is:[4]

```
X <y P-loc z>
```

The linking rules that relate the LCS to the predicate argument structure associate thematic roles (henceforth θ-roles) like *agent*, *theme*, and *recipient* with variables. An example of such a linking rule is:

[3] See Dorr (1987).

[4] This form of the predicate-argument structure is taken from Rappaport and Levin (1986). The variables outside the brackets are external arguments, and the variables inside the brackets are internal arguments. Henceforth, I will be representing such structures as annotated bracketed forms that correspond directly to tree structures; in this form external arguments correspond to positions outside the maximal projection of the verb, and internal arguments correspond to positions inside the verb's maximal projection.

```
Link the agent role with the external argument variable in the
predicate-argument structure.
```

The verb is then stored in the lexicon with its LCS and the θ-roles it assigns to the variables of the LCS:

```
PUT: (event (posit MOVE (thing X) (thing Y) (place Z)))
     X = agent, Y = theme, Z = locatum
```

The relations between the verb and its arguments are then manifested as grammatical functions in the syntactic underlying form of the sentence. The following illustrates an underlying form containing the verb *put*:

(1) $[_S\ [_{NP}\ \text{I}]\ [_{VP}\ \text{put}\ [_{NP}\ \text{the book}]\ [_{PP}\ \text{in the box}]]]$

2.2 Early Generation Designs: Direct Replacement

Several generation systems have used a direct replacement scheme (see Brown (1974), Forbus and Stevens (1981), Swartout (1981), and Winograd (1982)). Essentially, the technique involves templates that map an internal representation into surface text. As an example, we will look at the generation of text from internal concepts as found in Swartout's XPLAIN system (1981).

The XPLAIN phrase generator maps the concepts to phrases. For example, the concept:

```
((pvcs*f dangerous)*f (induced*o (by*o digitalis)))
```

is mapped to the phrase:

```
dangerous pvcs induced by digitalis
```

In order for this mapping to take place, a set of templates are used. The *tie* of each concept (indicated by a letter preceded by an asterisk (*)) points to the template that produces text for that concept. In the concept above, `*f` indicates that the second element in the list is a modifier of the first element; and `*o` indicates that the second element is the object of the first element. The template associated with `*f` places the second element (the modifier) before the first element if it is a single word or adjective; otherwise, the modifier is placed after the first element. Thus, `(pvcs*f dangerous)` is mapped to *dangerous pvcs*, whereas `(block*f (on*o the table))` is mapped to *block on the table*. The template associated with `*o` places the second element (the object) after the first element. Thus, `(by*o digitalis)` is mapped to *by digitalis*.

The advantage to using a direct replacement scheme is that expressions that are part of the (domain-dependent) internal representation for concepts can be mapped directly to surface text without the need of an underlying linguistic representation of the surface form. However, the disadvantages of the approach greatly outweigh this advantage. First, grammatical relations are identified by means of *ad hoc* rules that are implicit in the templates; not only are these rules language-specific, but they are too simplistic to handle sentences that

exhibit more complex phenomena (like raising and embedded sentences). Second, the system is not easily extendable, nor is its design readily transparent, because the rules mapping the internal representation to the surface form are entirely dependent on both the domain of the system and the language being generated.[5]

2.3 Syntactic Approach to Generation

As Chomsky's transformational paradigm quickly gained popularity in the 1960's, machine translation systems began to take a phrase structure approach to both parsing and generation. However, these systems were not based on a theory of *universal grammar* as is part of Chomsky's *Government-Binding* (henceforth GB) theory (see Chomsky (1981)). Rather than taking an approach that was oriented toward a syntactic *interlingua* (*i.e.*, language-independent form) based on deep structures, these systems used large language-specific grammars to parse and generate the source and target languages.

An example of a rule-based syntactic system is the generator of the METAL translation system (see Slocum (1984, 1985)), which is currently equipped with approximately 600 rules and 10,000 lexical entries in each of the two main languages (German and English). Bennett and Slocum (1985) argue that the *transfer* translation design (*i.e.*, the mapping of "shallow analyses of sentences" into "shallow analyses of equivalent sentences") is adequate for near-term applications. The argument against employing a "deep representation" is that long-term trials of such approaches seem to indicate that a suitable "deep representation" is not possible; furthermore, systems that use a "deep representation" cannot handle unrestricted input (some of which is ungrammatical).

Although a shallow analysis-synthesis of sentences might avoid some problems associated with current interlingual translation approaches, the complexity and language-specific nature of the rules translate into several problems. First of all, because the rules and lexical entries are so complex, the subject area must be very limited. Secondly, each rule is highly language-dependent in character; thus, there must be a set of target-specific transfer rules for every language that will serve as a target. This means that the rule system grows rapidly as each target language is added to the system. Thirdly, the rules are very stipulatory; there are no theoretical reasons for the rules being the way they are. Finally, each rule must carefully spell out the details of its application; thus, there is no way to capture linguistic generality among the rules in the system since general constraints are not factored out of the syntactic rules.

Two other systems that take a syntactic approach to generation are the TEXT system

[5] Admittedly, template-systems are generally not geared toward discovering or implementing a linguistic theory. Swartout acknowledges that his generator consists of the bare-minimum required to produce acceptable output; thus, linguistic principles are ignored:

> The generator should really be viewed more as an engineering effort that attempts to produce acceptable English rather than as a generation system that encodes deep linguistic principles. The main thrust of this thesis has been to investigate ways of representing the knowledge necessary to justify expert consulting systems. A generator is necessary to demonstrate the capabilities of the approach being espoused here, but the generator itself has not been the focus of the research.

```
(define-stylistic rule PREFER-ADJECTIVES-TO-NEW-SENTENCE
  ordering-on-attachment-points
   (attach-as-adjective attach-as-new-sentence)
  applicability-condition
   (if (includes-attachment-point 'attach-as-adjective
        usable-attachment-points)
       (not (or (will-be-complex-adjective-phrase
                 (usable-choices 'attach-as-adjective))
                (too-heavy-with-adjectives
                 (np-being-attached-to 'attach-as-adjective))))))
```

Figure 1: Stylistic Rule Used for Adjectival Attachment in MUMBLE

(McKeown (1983, 1985)) and the MUMBLE system (McDonald (1983, 1987)). These two systems are similar in that they use discourse and focus constraints to derive *messages* (*i.e.*, underlying representational forms) that are then used to generate syntactic structures corresponding to the surface text. Generation of syntactic structures in TEXT is based on the use of discrimination networks (to be described in section 4.3) and functional unification grammars (see Kay (1984)). Generation of syntactic structures in MUMBLE is based on the use of tree-adjoining grammars (see McDonald and Pustejovsky (1985b)) and stylistic rules (see McDonald and Pustejovsky (1985a)). Although both of these systems move away from the rule-based approaches of earlier schemes, they do not take advantage of structural and lexical generalization across languages. For example, the stylistic rules used for syntactic realization in MUMBLE are hand-generated; not only are they specific to English, but they are also specific to the domain of the system. Furthermore, they are often tedious to write, and their function in the system is generally not readily transparent. Figure 1 shows an example of such a stylistic rule. This rule is used for attachment of an adjective to a noun-phrase during the generation process.

In general, the move away from rule-based syntactic generation systems is a step in the right direction. However, care must be taken to prevent language-dependent devices from showing up in other forms. Language-independent universals need to be dealt with in a systematic way rather than in an *ad hoc* manner; language-specific idiosyncrasies can then be handled by a smaller set of individually applicable routines.

2.4 Semantic Approach to Generation

At the other end of the spectrum of generation systems are those systems which largely reject syntax as a basis of generation for language translation. Rather, generation is treated almost entirely on the basis of semantics, guided by a strong underlying model of the current situational context and expectations. (See Lytinen and Schank (1982), Lytinen (1985, 1987), Carbonell (1981), Cullingford (1986), Nirenburg *et. al.* (1985, 1986, 1987).)

The semantic-based (also called *knowledge-based* systems) are generally *interlingual*. That is, they employ a conceptual representation that is independent of any natural lan-

guage. Generally, this interlingua can be encoded by means of *primitive* meaning units. For example, in the MOPTRANS system (Lytinen and Schank, 1982), the Spanish word *capturar* is defined as GET-CONTROL in the dictionary. A *specialization* routine determines that *capturar* (= capture) is to be generated as the word *arrest* in the target language if the correct context (*e.g.*, a police search) has been instantiated.

Several arguments for choosing a semantic-based design over a syntactic-based design for generation systems have prevailed. The first is that the number of rules in a syntactic-based system would be enormous: a word may have several word senses, and each word sense would require a myriad of rules specifying the contexts in which the word sense might appear.

A second related problem is indexing. Since there are thousands of rules to choose from, "the amount of information the system would have to look for would be enormous, and deciding what information in the sentence was relevant for disambiguating the word in each particular context would be impossible."[6]

The third argument for a semantic-based design is that syntactic-based approaches tend to be overly concerned with the form of the input rather than the content (see Cullingford, 1986). Consequently, these *grammar-based* approaches do not easily handle deviant input (*e.g.*, input that is ungrammatical).

The claim that rule-based syntactic systems are both too large and too complex to adequately handle natural language translation may be well-grounded, but the semantic-based approach does not combat the problem! In attempting to tackle the problem of word disambiguation, semantic-based systems incorporate an incredibly massive amount of knowledge, effectively limiting the domain of subject matter.

An additional drawback to semantic-based approaches is that there is a loss of structure and style in generating the target text from the underlying (interlingual) form; consequently, the output of these systems is a paraphrase, not a translation. Although the *deep contextual meaning* of the input text is preserved, the emphasis or intent of the text is not always fully preserved. The claim is that any other system which attempts to preserve structure and style without the knowledge necessary for text understanding would often produce unreliable translations. However, the loss of structure and style may involve a loss of some of the meaning of the text. Most likely, the speaker chooses a particular structural realization in order to focus on a specific topic or to make a crucial point; the absence of structure preservation might result in a complete misinterpretation of the text.

Finally, another problem with knowledge-based generation systems is that they typically require an involved general inference mechanism in order to arrive at the surface form for a primitive concept. Rather than basing word selection on general lexical principles, complex inferencing routines are applied to conceptual representations. (Some examples of the type of inferencing that is required for lexical selection will be shown in section 4.3.)

2.5 The Shift Toward an LCS Generation Approach

The rule systems for existing natural language generators are still large, detailed, and complicated. Furthermore, generation systems lack linguistic motivation for the rules that they *do* have. The two primary tasks of natural language generation, *lexical selection* and *syn-*

[6]Lytinen and Schank, p. 13, 1982.

tactic realization, are not dealt with in a systematic manner; rather, *ad hoc* procedures are applied to underlying representations to arrive at surface structures.

If the basis of generation designs is shifted from complex, language-specific rules systems to modular syntactic theories that employ a well-defined lexical conceptual representation, several of the problems associated with earlier theories will be solved. Grammars will no longer be huge and complicated; small sets of lexical-semantic principles will replace complicated non-explanatory generation routines; and general inferencing will no longer be necessary. The next section describes the steps involved in constructing a generator on the basis of LCS.

3 Generation Scheme

Implicit in the generator are three components. The first two components are not the emphasis of this project, but they are nonetheless necessary. The first is a dictionary containing lexical conceptual representations that serve as the basis for generation of surface structures. Lexical items are stored in the dictionary with their associated properties, such as morphological feature sets, θ-roles that are assigned, and lexical-semantic representations. The second component, a syntactic and morphological synthesizer, maps a base form (henceforth called D-structure) to its corresponding surface form (henceforth called S-structure). The dictionary and synthesizer are standard components of any generation system; however, they differ from other systems in that they are based on LCS structures rather than rule-systems, semantic networks, or discrimination nets.

The final component of the generator is the emphasis of the project discussed here. This is the module that maps the lexical conceptual representation of a sentence to the D-structure of the target language sentence. This mapping requires both lexical replacement routines and linking rules in order to derive predicate-argument structures from the lexical-semantic representation. For example, in order to translate a source language verb like *gustar* to its target language equivalent *like*, lexical-replacement routines must match the LCS structures of these two verbs; then linking rules will be required in order to determine the structural positioning of the arguments (*e.g.*, that the *agent* is *externally* positioned in English, not *internally* as it is in Spanish).

Each of the three tasks of the scheme will be discussed in the following sections.

3.1 Construction of a Dictionary

The goals of this portion of the project are consistent with those put forth by the lexicon project in the Center for Cognitive Science at MIT. The focus is on representing knowledge of the syntactic and semantic properties of lexical items, particularly of verbs and their arguments.

In order to construct a dictionary, it is necessary to identify and utilize verbal properties through a study of lexical organization. Typically, lexical entries provide a minimal specification of the syntactic expression of the arguments of verbs. Within GB theory, there has been a move away from explicit use of subcategorization frames since the syntactic relations between the constituents in a sentence can be derived by two requirements: that θ-roles be

assigned under government and that nouns be assigned case to be well-formed.

In the process of building a dictionary, several subtasks are relevant:

1. Identification of thematic relations (and verification of their existence).

2. Organization of thematic relations (such as AGENT) into classes according to the constraints they are subject to. This may lead to the construction of a thematic hierarchy, if such a thing exists.

3. Construction of a mapping between thematic relations and syntactic arguments (taking into account the fact that the mapping may not be one-to-one).

4. Refinement of verb classes through examination of cooccurrence restrictions.

3.2 Mapping of D-Structure to S-Structure

Two components are required to map the D-structure to its corresponding surface form. The first is a syntactic synthesizer, essentially a movement module, that displaces tokens according to requirements of Case Theory (of Government-Binding). The second component is a morphological synthesizer that converts a root form and a set of features into a surface form.

The movement module accesses certain parameter settings corresponding to the language to be generated. These parameter settings determine the type of movement required. For example, the *wh*-movement parameter setting for English dictates that Subject-Aux Inversion (SAI) is to be triggered. Consequently, the generator will perform *wh*-movement and SAI to produce an output form. By contrast, in Spanish the *wh*-movement parameter is set such that V-Preposing (not SAI) occurs.

The morphological module converts `root+<feature>` forms into surface forms (*e.g.*, `"read+3S"` is converted to `"reads"`). This requires two mappings: one from features to possible affixes (*e.g.*, `"3S"` ⇒ `"s"`), and one from `<root>+<affix>` to possible surface forms (*e.g.*, `"read+s"` ⇒ `"reads"`).

3.3 Mapping of Lexical Conceptual Structure to D-Structure

In order to map the lexical conceptual representations that comprise a sentence to the target language D-structure of the sentence, two modules are needed: (1) a lexical replacement module that determines the corresponding target language (*e.g.*, English) words for the LCS's produced by parsing the source language (*e.g.*, Spanish) sentence; and (2) a syntactic module that performs the necessary operations in order to arrive at the target language D-structure of the sentence. Thus, there are two top-level operations during the mapping from LCS to D-structure: *selection* of target language words, and *linking* of surface-sentence words to their corresponding syntactic position.

The input to this component of the generator is a set of LCS's produced by a parser that maps source language sentences to their underlying structures. The output is the target language deep structure representation that will be used to generate the surface-sentence (by routines discussed in the last section). The selection of lexical translations for each token in a given underlying form begins with the predicate. The dictionary entry corresponding to the predicate is accessed, the surface verb of the sentence is selected, and the arguments of

the predicate are mapped to the case roles of the verb. Then the entries for each argument are accessed to return the lexical translations for the remainder of the proposition.

To illustrate this process, we will look at the translation of the word *poner* (= *put* in English). Suppose the source language sentence is *yo pongo el libro en la caja* (= *I put the book in the box*). First the lexical entry for the word *poner* is accessed. Recall that lexical entries contain the LCS and θ-marking requirements:

```
PONER: (event (posit MOVE (thing X) (thing Y) (place Z)))
       X = agent, Y = theme, Z = locatum
```

Next, the process of *selection* matches this LCS to that of the English verb *put* (repeated here for clarity):

```
(event (posit MOVE (thing X) (thing Y) (place Z)))
```

The deep structure of the sentence is dependent on its verb (*e.g.*, how many objects it takes, whether it has a subject, *etc.*). Once a verb has been selected to translate the predicate, the semantic arguments of the deep structure are filled with the instantiated arguments of the predicate. Thus, after *put* lexically replaces *poner*, the *linking* process is activated. The θ-marking properties of *put* combined with linking rules for *agent*, *theme*, and *locatum* derive the following predicate-argument (deep) structure:

(2) $[_S\ [_{NP}\ \text{I}]\ [_{VP}\ \text{put}\ [_{NP}\ \text{the book}]\ [_{PP}\ \text{in the box}]]]$

Here, the linking rules have mapped the *agent* (= *I*) into external argument position, and the *theme* (= *the book*) and *locatum* (= *in the box*) into internal argument position. (See section 4.3 for an example of a linking rule.)

4 Embodiment of Linguistic Theory

The above scheme of representation and generation should be constructed in such a way that properties that are shared among all languages are handled by a unified set of "core" linguistic principles, while the differences among languages are accounted for by a set of possible parameters of variation. In this view, many properties of particular languages can be accounted for through the interaction of principle-based subsystems, while complexes of properties differentiating otherwise similar languages should (ideally) be reducible to a single parameter, fixed in one or another way.[7] Thus, in order to build a generator for machine translation, it is necessary to determine both the lexical properties that make words similar across languages, as well as the properties that distinguish words cross-linguistically. In terms of the generation approach discussed here, the "core" linguistic principles are those procedures required for selection of words and linking of LCS to syntactic structure, while the parameterization occurs in the lexicon, with individual lexical items taking on their own language-particular "meaning" and thematic role-assigning properties.

[7] A brief overview of the principles of GB-theory is presented in Dorr, 1987.

Recall that there are two top-level operations for mapping LCS to D-structure: *selection* and *linking*. Before developing procedures for these two operations, it is necessary to examine some examples of source-to-target language translations and to determine some of the difficulties that might arise during generation of target language sentences. Of particular concern are the problems of thematic divergence (which makes selection difficult) and structural divergence (which makes linking difficult). Some examples should shed some light on what is needed for both the lexical conceptual representation of words as well as the mapping from this representation to the surface form. In the examples shown here, Spanish and English are the two languages used. Other languages (*e.g.*, German and Japanese) will also be tested when the implementation is complete. A generation scheme based on LCS will be presented as a solution for the problems exhibited in the examples. I will then discuss the goals of the scheme.

4.1 Example 1: Thematic Divergence as a Problem for Lexical Selection

The task of lexical selection is difficult because of the possibility of *thematic divergence*, *i.e.*, a difference in the order of thematic role assignment. An example of thematic divergence is the translation of the Spanish word *gustar* to the English word *like*. Although these two verbs are semantically equivalent, their argument structures are not identical: the subject of *gustar* is the *patient* of the action, whereas the subject of *like* is the *agent* of the action. Thus, we have:

(3) Me gusta el libro a mí
(To me the book pleases me)
'I like the book'

In general, cases such as (3) are not problematic. The difference in order of thematic role assignment is easily manipulated by simple procedures that check thematic requirements of the two verbs. Furthermore, the verb *gustar* can have the translation *like* stored directly in its lexical entry since this is the only possible translation for it. However, problems arise when a verb has more than one translation depending on the selectional restrictions of its arguments. Two examples are the English words *slash* and *smear*:

(4) (i) He slashed the woman
'Dio cuchilladas a la mujer'

(ii) He slashed the paper
'Cortó el papel'

(5) (i) She smeared her makeup
'Embarró su maquillaje'

(ii) She smeared the wall with paint
'Pintarrajeó a la pared'

In (4)(i) the translation of *slash* is the composite form *dar cuchilladas a*, whereas in (4)(ii) the translation of *slash* is the single word *cortar*. In (5)(i) *smear* is translated directly to the

Spanish word *embarrar*, but in (5)(ii) *smear* translates to the more complex Spanish word *pintarrajear a* which implicitly incorporates the nominal argument *paint*. In such cases, it is not possible to store direct translations in the lexical entries of the verbs since argument incorporation is sometimes required: in the case of *slash*, the translation breaks down into a more basic argument-structure *dar cuchilladas a* (literally translated, it means *to give knife-wounds to*); and in the case of *smear* the translation combines the argument *paint* with the verb in order to arrive at *pintarrajear a*. Thus, we see a need for word definition in terms of more basic meaning structures in order to choose an accurate translation at generation time.

4.2 Example 2: Structural Divergence as a Problem for Linking

Linking a meaning structure to its surface-syntactic representation is difficult because of cases of structural divergence between languages. In general, in these cases, there is also a selection problem (in fact, the choice of an equivalent target language verb may lead to a non-equivalent surface-structure representation). An example of structural divergence is the translation of the Spanish verb *tener* as the English verb *be* in certain cases:

(6) Tengo calor
(I have heat)
'I am hot'

The predicate-argument structure for *tener calor* is:

(7) $[_{VP}$ tener $[_{NP}$ calor$]]$

The predicate-argument structure for *be hot* is:

(8) $[_{VP}$ be $[_{AP}$ hot$]]$

Here, a noun-phrase argument must be changed into its adjectival-phrase counterpart.

There are also structural divergences in which adjuncts and arguments are either added or deleted in the resulting translation. In general, if a token is added, that token was implicit in the original source language verb; if a token is deleted, that token becomes incorporated into the target language verb. An example in which a token is implicit in the source language verb is the composite verb *throw down*; the translation is *echar por tierra* (literally, throw to the ground):

(9) He threw down the book
'Echó por tierra el libro'

Whereas *throw down* is syntactically a single unit, *echar por tierra* consists of a verb with a prepositional adjunct:

(10) $[_{VP}$ $[_{V}$ throw-away$]]$
$[_{VP}$ $[_{VP}$ $[_{V}$ echar$]]$ $[_{PP}$ por tierra$]]$

On the other hand, the composite verb *throw away* is simply translated as *tirar*, which has the token *away* incorporated directly:

(11) He threw away the book
'Tiró el libro'

Thus, the two syntactic structures are essentially the same:

(12) $[_{VP}\ [_{V}$ throw-down$]]$
$[_{VP}\ [_{V}$ tirar$]]$

Similarly, the Spanish verb *forzar* may have the translation *break into* (as in (13)(i)) if the token *la entrada* is present (the literal translation is *force the entry*), or it may be translated simply as *force* (as in (13)(ii)):

(13) (i) Forzó la entrada a la casa
'He broke into the house'

(ii) Forzó el ejército rendir
'He forced the army to surrender'

The corresponding divergent and equivalent structures for these examples are:

(14) (i) $[_{VP}\ [_{V}$ forzar$]\ [_{NP}$ la entrada$]]$
$[_{VP}\ [_{V}$ break$]\ [_{PP}$ into $\cdots]]$

(ii) $[_{VP}\ [_{V}$ forzar$]\ [_{NP}$ el ejército$]]$
$[_{VP}\ [_{V}$ break$]\ [_{NP}$ the army$]]$

4.3 Lexical Conceptual Structure

The translation examples above provide strong evidence that a suitable representation for lexical conceptual structure is needed. Previously, generation systems used discrimination nets in order to select the appropriate surface forms for underlying concepts. For example, in Carbonell, *et. al.* (1981) the sentence *Mary hit John* is represented as:

```
(event EV001
   (action PROPEL)
   (agent MARY)
   (object JOHN)
   (instrument *UNKNOWN*)
   (force *ABOVE-AVERAGE*)
   (intentionality *POSITIVE*))
```

In order to translate the above concept into Spanish, the main action (PROPEL) is mapped to the discrimination network shown in figure 2. This network is then used to choose the correct verb. As a series of If-Then statements, this net expands into the complex block of code shown in figure 3.

In the above scheme, there is no representation of the "meaning" of the verbs being generated; rather, the mapping from concept to surface form is performed by means of *ad hoc* inferencing procedures that test selectional restrictions of arguments and act accordingly. The problem with such an approach is that the network can grow very large as more verbs are

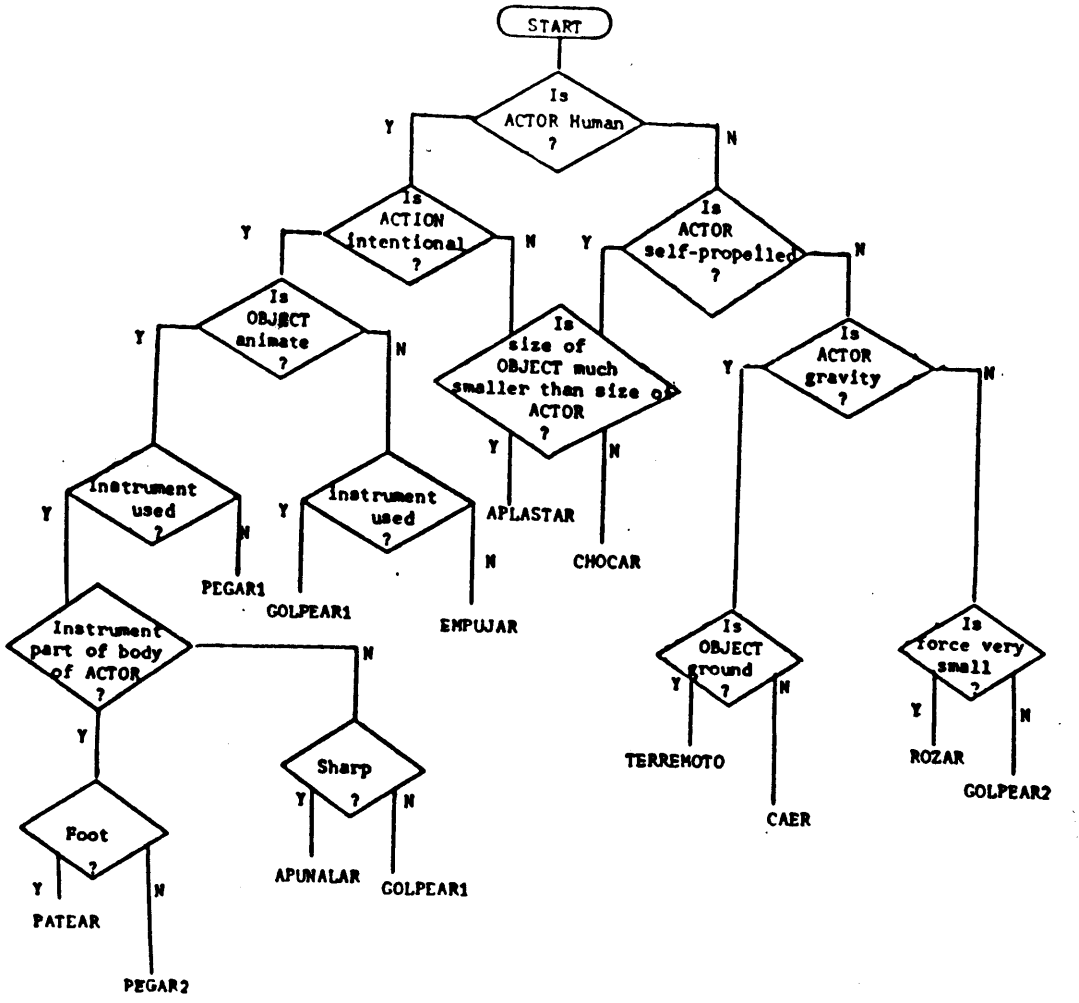

Figure 2: Discrimination Network for PROPEL, from Carbonell, *et. al.* (1981)

added, and the time it takes to search for an appropriate verb is exponential. Additionally, this same network must be searched *every time* a verb expressing PROPEL conceptualizations is to be generated; thus, a great deal of time is wasted testing for irrelevant selectional restrictions. For example, even though no selectional restrictions apply on the arguments of "golpear2", the entire network must still be searched in order to generate this surface form.

There are additional problems with this network approach. First, the network structure does not readily accommodate adding new words or deleting old ones since major surgery of the network is typically required. Second, generation of compound predicate-argument structures is not accommodated by such a scheme since all the paths lead to a single lexical item (not a *set* of lexical items); thus, neither thematic nor structural divergence is explained in such a scheme. Finally, the scheme is not general enough to apply across several languages since surface forms are not represented as basic units of meaning; consequently, a new network must be hand-written for each language to be generated.

```
If ACTOR is Human
  Then If ACTION is intentional
          Then If OBJECT is animate
                  Then If INSTRUMENT is used
                          Then If INSTRUMENT is part of body of ACTOR
                                  Then If foot
                                          Then "patear"
                                          Else "pegar2"
                                  Else If sharp
                                          Then "apuñalar"
                                          Else "golpear1"
                          Else "pegar1"
                  Else If INSTRUMENT is used
                          Then "golpear1"
                          Else "empujar"
          Else If size(OBJECT) < size(ACTOR)
                  Then "aplastar"
                  Else "chocar"
  Else If ACTOR is self-propelled
          Then If size(OBJECT) < size(ACTOR)
                  Then "aplastar"
                  Else "chocar"
          Else If ACTOR is gravity
                  Then If OBJECT is ground
                          Then "terremoto"
                          Else "caer"
                  Then If force is very small
                          Then "rozar"
                          Else "golpear2"
```

Figure 3: If-Then Code for PROPEL Discrimination Network

In order to model cross-linguistic variations in predicate-argument structures such as these, an adequate lexical-semantic representation is required. According to Talmy (1985) (following Jackendoff and Gruber), verbs should be defined in terms of many semantic components that contribute to the overall meaning. Thus, verbs may have a semantic representation that is not entirely exhibited at the level of syntactic structure. For example, the verb *enter* incorporates an "understood" particle *into* as part of its meaning structure; this particle manifests itself in the equivalent composite predicate *go into*. This *incorporation* or *conflation* of properties is where cross-linguistic parametric variations are revealed. For example, where English conflates manner and motion in *the boat floated on the water*; Spanish disallows this conflation, requiring a syntactic realization for each semantic component: *la barca se mudaba flotando en el agua* (literally, this is *the boat moved floating on the water*).

Using a representation similar to that of Jackendoff (1972, 1983), the semantic equivalence between *enter* and *go into* is easily modeled:

```
enter = (event (posit GO (thing X) (path TO (place IN (thing Z)))))
go = (event (posit GO (thing X) (path Y)))
into = (path TO (place IN (thing Z)))
```

Here, the LCS forms for *go* and *into* can be composed into the more complex LCS form for *enter*.

Similarly, the LCS for *float* can be decomposed into the LCS forms for *move* and *float*:

```
float = (event (posit GO (thing X) (property BUOYANT) (path Z)))
move = (event (posit GO (thing X) (property Y) (path Z)))
floating = (property BUOYANT)
```

Note that this approach differs from that of Carbonell, *et. al.* (1981) in that the "meaning" structure (*i.e.*, LCS) is stored directly in the lexical entry of each word; it is not derived by network navigation. The primitives of the scheme described here are used compositionally to define words of the source and target languages. Because source language definitions are matched against target language definitions to select the correct target language words, there is no need to test properties of arguments; thus, time-consuming and unnecessary searches are avoided. Furthermore, through a combination of a small set of linking rules and a list of θ-role assigning properties, the LCS scheme provides a facility for syntactic realization of surface forms. In contrast, it is not clear how syntactic structure is realized using the discrimination network approach.

With respect to generation in the context of machine translation, this decomposition of meaning is useful in the mapping from underlying LCS forms to target language surface forms. In dealing with thematic divergence, LCS's provide a uniform representation for equivalent source-target pairs. Thus, both *gustar* and *like* have the following LCS:

```
(event CAUSE (thing X) (poss BE (thing Y) (property PLEASED)))
```

The difference in thematic role assignment can then be determined by means of properties of the individual lexical items. The *agent* θ-role will be assigned to `X` in the case of *gustar* and to `Y` in the case of *like*.

The LCS scheme is also handy in the case where a verb may translate into more than one surface form depending on its arguments. We saw that *smear* translates either directly as *embarrar* or as the conflated verb *pintarrajear* (if the object that is being smeared is *paint*). The LCS for *smear* and *embarrar* is:[8]

```
(event (poss MOVE (thing X (property FLUID))
                  (path ALONG (place (poss ON) (thing Y)))))
```

The representation for the noun *paint* is:

```
(thing PAINT (property FLUID))
```

and the representation for *pintarrajear* is:

```
(event (poss MOVE (thing PAINT (property FLUID))
                  (path ALONG (place (poss ON) (thing Y)))))
```

Thus, *smear paint* will be translated as *pintarrajear* since the LCS of the noun *paint* matches the object of MOVE in the LCS for *pintarrajear*. On the other hand, *smear makeup* will be translated as *embarrar maquillaje* since the word *maquillaje* does not match the object of MOVE in the LCS for *pintarrajear*.

The LCS scheme also provides an adequate model of structural divergence in the linking of meaning structure to its surface-syntactic representation. Recall that *tener calor* is translated to the structurally divergent form *be hot*. The LCS for *have* is:

```
(state (poss BE (thing X) (place (poss AT) (thing Y))))
```

The LCS for *be* is:

```
(state (ident BE (thing Y) (property X)))
```

In the case of *tener calor*, the first LCS is instantiated; thus, `X` is set to be *calor* and `Y` is set to be the *agent* of the predicate. (The assignment of *agent* to `Y` is specified as a language-particular property of the verb *tener* in the lexicon.) This LCS is then mapped to the LCS for *be*, where `Y` is the *agent*, and `X` is converted into the property *hot* corresponding to *calor* (*i.e.*, the nominal form *heat* is changed into the adjectival form *hot*). Note that the difference between the source and target structure is determined solely on the basis of the identification of `X` as a *property* rather than a *thing* in the LCS for the target language verb.

Structural divergence due to conflation is also modeled by the LCS scheme. As we have seen, the composite verb *break into* is translated as *forzar la entrada*. The LCS for *break into* is:

[8]The representation shown here is primarily based on Jackendoff's conceptual structures; however, discussion with Michael Brent influenced me to add the property FLUID. A more elaborate LCS form for verbs such as *throw*, *smear*, and *spray* are in Brent (1988); however, the simple representations shown here are adequate for the purposes of this discussion.

```
(event GO (thing X) (path TO (place (poss IN) (thing Y)))
                    (property VIOLENTLY))
```

The LCS for *forzar* is:

```
(event GO (thing X) (path Y) (property VIOLENTLY))
```

and the LCS for *entrada* is:

```
(path TO (place (poss IN) (thing Y)))
```

Thus, in linking the compound LCS for *break into* to the target language syntactic form, the compound LCS must be decomposed into the individual LCS's for *forzar* and *entrada*; these decomposed structures are then linked to the surface-syntactic representation for *forzar la entrada*.

4.4 Goals of the Generation Scheme

If the system is to handle the examples mentioned above, it should embody modern linguistic theory so that it provides an explanatory model of language generation. In order to be explanatorily adequate, the system must base its operation on general procedures that adhere to well-defined linguistic principles. Furthermore, the system must include several parameters of variation so that it is flexible enough to handle several languages. This parameterization also fulfills the goal of extendability; adding new languages reduces to changing parameter values of the system.

An additional goal is that of expressive power. The primitives that are the basis of the system should be designed with cross-linguistic applicability in mind. In order to parameterize the system, the primitives must be adequate for composition into complex meaning structures that map into the words of both the source and the target language.

The goal of avoiding *ad hoc* rules can be fulfilled if the scheme makes use of a more restrictive theory of lexical semantics than that of existing generation systems. Furthermore, the semantic structures should be designed so that general inference will not be required in order to select target language words in the generation process. As long as the mapping from LCS to surface form is uniform across all LCS forms, general inferencing procedures will not be required. The operations of lexical selection and syntactic realization are simplified once rules and general inferencing are eliminated: LCS and θ-role mappings obviate the need for complicated network searches and rule applications. Finally, exponential search or varying search time for different words can be avoided if there are no general inference procedures.

An example of how the LCS-based translation process will operate at each stage is the following:

Source Language Sentence:

El libro me gusta a mí

Source Language Parse:

$[_S\ [_{NP}$ el libro$]\ [_{VP}$ me gusta $[_{NP}$ a mí$]]]$

Instantiation of LCS (Spanish):

(event CAUSE (thing <libro>$_{agent}$) (poss BE (thing <mí>$_{goal}$) (property PLEASED)))

Instantiation of LCS (English):
(event CAUSE (thing <book>$_{goal}$) (poss BE (thing <I>$_{agent}$) (property PLEASED)))
Target Language Generation:
[$_{S}$ [$_{NP}$ I] [$_{VP}$ like [$_{NP}$ the book]]]
Target Language Sentence:
I like the book.

5 Work To Be Done

The generator will be an "inverse UNITRAN parser;" it will replace the generation component that is currently part of the UNITRAN machine translation system. In order to build the LCS-based generator, several tasks must be undertaken. First, the selection of primitives is necessary. All of the LCS forms are based on cross-linguistically applicable primitives (like GO and BE) that must be carefully defined. The primitives must be designed so that they are easily programmable, but they are not decomposable (in any language).

The next task is the construction of the LCS forms. This means that the primitives must be composed in a certain manner in order to arrive at certain meaning structures. Section 4.3 gives some examples of how the primitives (like GO, BE, *etc.*) are composed to form words with complex meanings structure (like *enter* and *go into*).

An additional task is to provide a mapping from LCS to surface structure. This includes routines for both selection and syntactic realization. Principles that are already built into the UNITRAN system will be operative during this mapping (as they are during parsing); however, thematic role assignment will have to be extended to include assignment to variables in LCS.

In addition to the actual construction of the system, methods of testing and evaluating the system need to be devised. In particular, cross-linguistic generalization will need to be tested. This can be done by trying the system on other languages. In addition to English and Spanish, the two languages that will be tested are German and Japanese. It must be possible to perform lexical selection on the basis of LCS structures for all four of these languages; furthermore, syntactic realization must work correctly for each language. In order for this endeavor to be realized, parameters of variation must be established. On the syntactic side, the UNITRAN system is already parameterized according to GB theory. On the lexical-semantic side, parameterization occurs in the lexicon and in the linking rules. Once the settings are established for the languages handled by the system, an evaluation can be made on the basis of the correctness of translated sentences.

6 Difficulties to be Addressed

The first consideration in building the generator is that it must be constructed so that it is based on the same principles that the parser uses. The principles that are already part of UNITRAN are primarily syntactic in nature; thus, they will not affect the lexical conceptual structure, but they will affect how the syntactic portion of the generator operates. For example, during the structural realization process (or *linking*), the satisfaction of certain syntactic constraints must still be maintained (*e.g.*, that a verb governs its object in English).

Another difficulty is the construction of primitives. It is not clear how many primitives to have, nor is it easy to determine that the primitives are indeed non-decomposable in every language. Furthermore, the primitives must be easy to represent and to compose into complex meaning structures.

The process of lexical selection might also be problematic in that it is not always easy to determine how far an LCS should be broken down before generating a surface form. For example, recall that the LCS for *smear paint* is:

```
(event (poss MOVE (thing PAINT (property FLUID))
                  (path ALONG (place (poss ON) (thing Y)))))
```

In Spanish, this can either be broken down into two non-composite surface forms *embarrar pintura* (literally, *smear paint*), or it can be left as the composite surface form *pintarrajear*. In order to solve this problem, a principle of conservation will be needed: the most complex set of words that matches an LCS will be chosen for generating a surface form. In the above example, *pintarrajear* is chosen.

Another difficulty to be addressed is that thematic role assignment will need to be modified to apply to instantiated LCS arguments, but it still must remain consistent with syntactic principles that are already part of the system (*e.g.*, the θ-Criterion). Thus, while thematic roles are used in the mapping from the LCS to the syntactic structure, they must still be preserved after the syntactic structure is derived in order to satisfy syntactic principles that already exist.

A final difficulty to be addressed is that of final realization of the source language surface structure. Once the appropriate LCS has been chosen, the correct surface forms have been selected, and linking has taken place to derive a syntactic structure, the generator must perform certain movement operations in order to arrive at the final surface structure. For example, the V-Preposing operation in Spanish fronts a verb when a *wh*-question is asked:

(15) ¿Qué vio Juan?
(What saw John?)
'What did John see'

In order to generate the V-preposed form, a movement parameter must be accessed. This parameter is set to V-prepose in Spanish (and SAI in English); thus, V-preposing will occur in Spanish (and SAI in English) when a *wh*-phrase is found in the correct position.

Despite these difficulties, once the generator design is chosen, it should be possible to make headway toward reducing the amount of information and time required for machine translation. Ideally, the system should contain a small and tightly constrained set of parameterized principles.

7 References

Barton, G. Edward, Jr. (1984) "Toward a Principle-Based Parser," Massachusetts Institute of Technology, Cambridge, MA, AI Memo 788.

Bennett, Winfield S. and Jonathan Slocum (1985) "The LRC Machine Translation System," *Computational Linguistics* 11:2-3, 111-121.

Berwick, Robert C. (1987) "Principle-Based Parsing," Massachusetts Institute of Technology, Cambridge, MA, AI Technical Report 972.

Brent, Michael R. (1988) "Decompositional Semantics and Argument Expression in Natural Language," Master of Science thesis, Massachusetts Institute of Technology.

Brown, Gretchen (1974) "Some Problems in German to English Machine Translation," Massachusetts Institute of Technology, Cambridge, MA, MAC Technical Report 142.

Carbonell, Jaime G., Richard E. Cullingford, and Anatole V. Gershman (1981) "Steps Toward Knowledge-Based Machine Translation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-3:4.

Chomsky, Noam A. (1981) *Lectures on Government and Binding*, Foris Publications, Dordrecht.

Chomsky, Noam A. (1986) *Knowledge of Language: Its Nature, Origin and Use*, MIT Press, Cambridge, MA.

Cullingford, Richard E. (1986) *Natural Language Processing: A Knowledge-Engineering Approach*, Rowman and Littlefield, Totowa, New Jersey.

Dorr, Bonnie J. (1987) "UNITRAN: A Principle-Based Approach to Machine Translation," AI Technical Report 1000, Master of Science thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Forbus, K., and A Stevens (1981) "Using Qualitative Simulation to Generate Explanations," *Proceedings of Third Annual Conference of Cognitive Science*, Berkeley, CA.

Gruber, J. S. (1965) "Studies in Lexical Relations," Ph.D. thesis, Department of Information Science Department, Massachusetts Institute of Technology.

Hale, Kenneth and M. Laughren (1983) "Warlpiri Lexicon Project: Warlpiri Dictionary Entries," Massachusetts Institute of Technology, Cambridge, MA., Warlpiri Lexicon Project.

Hale, Kenneth and Jay Keyser (1986) "Some Transitivity Alternations in English," Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA., Lexicon Project Working Papers #7.

Jackendoff, Ray S. (1972) *Semantic Interpretation in Generative Grammar*, MIT Press, Cambridge, MA.

Jackendoff, Ray S. (1983) *Semantics and Cognition*, MIT Press, Cambridge, MA.

Kay, Martin (1984) "Functional Unification Grammar: A Formalism For Machine Translation," *Proceedings of COLING-84*, Stanford, CA, 75–78.

Levin, Beth (1977) "Mapping Sentences to Case Frames," Massachusetts Institute of Technology, Cambridge, MA, AI Working Paper 143.

Lytinen, Steven and Roger Schank (1982) "Representation and Translation," Department of Computer Science, Yale University, New Haven, CT, Technical Report 234.

Lytinen, Steven L. (1985) "Integrating Syntax and Semantics," *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, 167–177.

Lytinen, Steven L. (1987) "Integrating Syntax and Semantics," in *Machine Translation: Theoretical and Methodological Issues*, Sergei Nirenburg (eds.), Cambridge University Press, Cambridge.

McDonald, David D. (1983) "Natural Language Generation as a Computational Problem," in *Computational Models of Discourse*, Brady, Michael and Robert C. Berwick (eds.), MIT Press, Cambridge, MA.

McDonald, D. D., and J. Pustejovsky (1985a) "A Computational Theory of Prose Style for Natural Language Generation," *Proceedings of the Second Conference of the European Chapter of the Association for Computational Linguistics*, University of Geneva, Geneva Switzerland, 187–193.

McDonald, D. D., and J. Pustejovsky (1985b) "TAGs as a Grammatical Formalism for Generation," *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, IL, 94–103.

McDonald, David D. (1987) "Natural Language Generation: Complexities and Techniques," in *Machine Translation: Theoretical and Methodological Issues*, Sergei Nirenburg (eds.), Cambridge University Press, Cambridge.

McKeown, Kathleen (1983) "Focus Constraints on Language Generation," *Proceedings of the Third International Joint Conference of Artificial Intelligence*, Karlsruhe, 582–587.

McKeown, Kathleen (1985) *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Cambridge University Press, Cambridge.

Nirenburg, Sergei, Victor Raskin, and Allen B. Tucker (1985) "Interlingua Design for TRANSLATOR," *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, 224–244.

Nirenburg, Sergei (1986) "Machine Translation," *Presented as a Tutorial at the 24th Annual Meeting of the Association for Computational Linguistics*, Columbia University, New York City, New York.

Nirenburg, Sergei, Victor Raskin, and Allen B. Tucker (1987) "The Structure of Interlingua in TRANSLATOR," in *Machine Translation: Theoretical and Methodological Issues*, Sergei Nirenburg (eds.), Cambridge University Press, Cambridge.

Rappaport, Malka, and Beth Levin (1986) "What to Do with Theta-Roles," Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA., Lexicon Project Working Papers #11.

Slocum, Jonathan (1984) "METAL: The LRC Machine Translation System," Linguistics Research Center, University of Texas, Austin, Working Paper LRC-84-2.

Slocum, Jonathan (1985) "A Survey of Machine Translation: Its History, Current Status, and Future Prospects," *Computational Linguistics* 11:1, 1–17.

Swartout, William (1981) "Producing Explanations and Justifications of Expert Consulting Programs," Massachusetts Institute of Technology, Cambridge, MA, LCS Technical Report 251.

Talmy, L. (1985) "Lexicalization Patterns: Semantic Structure in Lexical Forms," in *Grammatical Categories and the Lexicon*, T. Shoper (eds.), Cambridge University Press, Cambridge.